BioEL: A Comprehensive Python Package for Biomedical Entity Linking

Prasanth Bathala* **Christophe Ye*** **Batuhan Nursal***
Shubham Lohiya* **David Kartchner** **Cassie S. Mitchell**

Georgia Institute of Technology

{pbathala3, cye73, bnursal3, slohiya3, dkartchner3}@gatech.edu,
cassie.mitchell@bme.gatech.edu

Abstract

Entity Linking in biomedical literature is a critical task that enhances the extraction and integration of information from diverse scientific literature. This paper introduces "**BioEL**", a robust open-source Python package developed to advance **Bio**medical **E**ntity **L**inking. BioEL serves as an accessible and comprehensive tool aimed at researchers and practitioners, facilitating the implementation and comparison of BioEL tasks. The package encompasses four key components: (1) **Ontology Object**, which manages and applies ontologies across datasets; (2) **Dataset Object**, integrated with BigBIO for handling annotated corpora; (3) **BioEL Model Object**, supporting training of various BioEL models, including those with pre-trained weights; and (4) **Evaluation Framework**, offering robust metrics and methodologies for assessing model performance. The library is extensible and configurable, fostering ongoing development and customization opportunities. This paper details the design principles, core components, and functionalities, and presents benchmarking results on entity-linking tasks using prevalent biomedical datasets.

Our unified evaluation framework and all included models are on GitHub at https://github.com/pathology-dynamics/biomedical-entity-linking.

1 Introduction

Entity Linking in biomedical literature is necessary to harness the vast amount of information embedded within scientific texts. The task involves linking mentions of entities such as genes, proteins, and diseases to their corresponding entries in the Knowledge Base (KB) so researchers can navigate, integrate, and analyze biomedical knowledge.

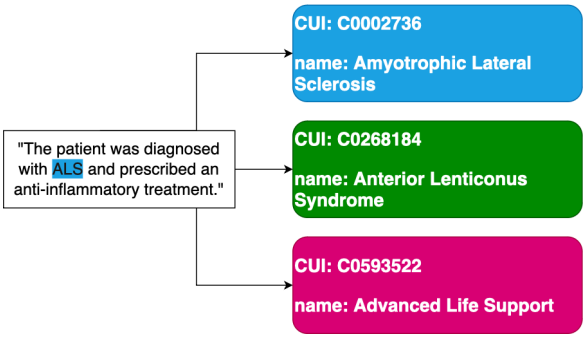

Figure 1: Disambiguation challenge in Biomedical Entity Linking

Despite its significance, biomedical entity linking faces major challenges due to complex terminology, limited labeled data, and a lack of interoperability between ontologies. Kartchner and colleagues (Kartchner et al., 2023) introduced a unified evaluation framework for state-of-the-art Biomedical Entity Linking models. However, accessing, training and evaluating models across various tasks and datasets still requires extensive domain knowledge and time. The manual manipulation of different files, model versions, model settings, and data preprocessing steps decreases reproducibility and slows innovation:

- *Lack of model interface standardization* - Models have different requirements for input formats, hyperparameters, and outputs, which complicates the integration and comparison of multiple models. For example, arboEL (Agarwal et al., 2022) supports only MedMentions, BC5CDR, and ZeShEL datasets, and each has its own preprocessing script.
- *Dataset and ontology format variance* - Datasets and ontologies vary in formats and annotation standards, resulting in significant effort to convert and standardize. For example, MEDIC (Davis et al., 2019) and Entrez

* These authors contributed equally to this work.

Findings of the Association for Computational Linguistics:
NAACL 2025, pages 1709–1721

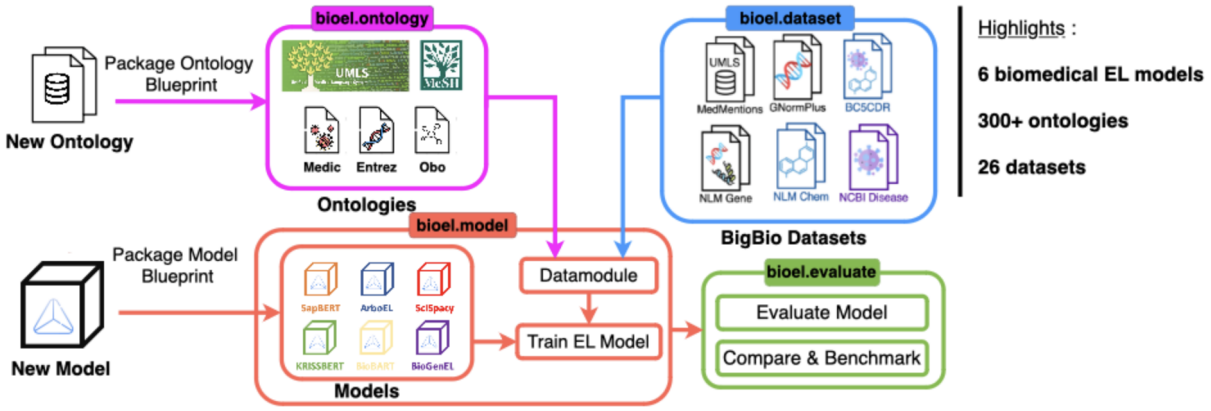

Figure 2: Overview Structure of the Package.

Gene (Maglott et al., 2005) have distinct structures and contain different types of information in their ontologies.

- *Inconsistent task evaluations across the literature* - Studies employ different metrics and evaluation protocols, which makes it challenging to reproduce results and objectively compare model performance. For instance, (Zhang et al., 2022) showed that BYOSIN (Sung et al., 2020) and SapBERT (Liu et al., 2021) do not resolve ambiguity for an entity mention that matches multiple entities. The prediction is considered correct if the gold entity is among the matching entities. Additionally, some models, like arboEL (Agarwal et al., 2022), report results up to recall@64, while others, like KRISSBERT (Zhang et al., 2022), only report accuracy.

- *Lack of model and dataset documentation* - Each model needs instructions for testing on different datasets. Lack of instructions or having poor or outdated documentation has been a large impediment in reproducing published experiments. MedLinker (Loureiro and Jorge, 2020), BioGenEL (Yuan et al., 2022b), and ArboEL were previously found to be the most challenging to adapt and reproduce (Kartchner et al., 2023).

- *Inefficient handling and processing* - Machine Learning researchers often repeat data preprocessing, model training, and evaluation setup for each model and dataset, making it a time-consuming and inefficient process.

In this work, we introduce **BioEL**, an open-source Python package tailored for biomedical entity linking. BioEL establishes a standardized and user-friendly framework that streamlines the execution and assessment of biomedical entity linking tasks. By integrating existing resources and methodologies, this package makes several key contributions:

- BioEL currently supports **6 biomedical entity linking models**, utilizing an extensive collection of **over 300 ontologies** and spanning **26 diverse datasets**.

- BioEL offers a unified interface for models, datasets, and evaluation metrics, which significantly reduces setup time and effort in biomedical entity-linking research.

- The package promotes consistent model evaluation by facilitating objective comparisons between different approaches and enhancing result reproducibility.

- The BioEL user-friendly design and comprehensive documentation make the package accessible to new and existing practitioners.

- BioEL serves as a robust platform for developing and testing new models, thereby accelerating advancements in the field.

- BioEL is designed to be extensible and configurable - facilitating seamless integration of new models into the framework.

This paper outlines the design and implementation of BioEL and showcases its efficacy through case studies and evaluations. In summary, the open-source BioEL Python package enhances the efficiency, accessibility, and impact of biomedical entity linking research.

2 Related work

Open-source packages such as BLINK (Wu et al., 2020) and SpaCy (Honnibal and Montani, 2017) are commonly used for entity linking and are easy to use. However, they are general-purpose tools that lack dedicated support for biomedical data and each relies on its own model.

BioDBLinker (Walsh et al., 2020) is a rule-based library designed for linking entities across biological knowledge bases, enhancing data integration and interoperability. However, it does not perform traditional entity linking, such as resolving mentions to specific entities within a knowledge base, nor does it support a unified entity linking model.

BENT: Biomedical Entity Annotator (Ruas and Couto, 2022) is an entity linking tool that provides end-to-end named entity recognition and linking in the biomedical domain. It utilizes a graph-based approach tightly integrated with 14 supported ontologies and employs pre-processing dictionaries to map recognized entities to ontology entries. However, the package is restricted to this methodology and does not support alternative models or linking techniques.

BERN2 (Sung et al., 2022) and HunFlair2 (Sänger et al., 2024) also follow a fixed methodology for entity linking, combining rule-based techniques with BioSyn (Sung et al., 2020) and SapBERT (Liu et al., 2021). However, like BENT, they lack support for diverse and dataset independent entity linking models, limiting their adaptability to broader applications. Furthermore, both tools provide only pre-trained models optimized for specific biomedical entity types—genes/proteins, diseases, chemicals, species, and cell lines—while entity linking for unsupported types relies solely on rule-based methods.

Table 1 provides a comparison of BioEL's functionalities with BENT, BERN2 and HunFlair2

Feature	BENT	BERN2	HunFlair2	BioEL
Dataset Loader	✗	✗	✗	✓
Ontology Support	✓	✓	✓	✓
Training Pipeline	✗	✗	✗	✓
Evaluator Object	✗	✗	✓	✓
Benchmarking framework	✗	✗	✓	✓
Plotting & Visualization	✗	✗	✗	✓
Pre-trained model	✓	✓	✓	✓

Table 1: Comparison of BioEL with BENT (Ruas and Couto, 2022), BERN2 (Sung et al., 2022) and HunFlair2 (Sänger et al., 2024)

3 Architecture and Overview

3.1 Architecture

BioEL is a unified environment that integrates core components to streamline biomedical entity linking model development and evaluation without technical inconsistencies or redundant efforts.

- **Ontology** - `bioel.ontology` handles and utilizes domain-specific knowledge representations, enabling the use of established ontologies like Unified Medical Language System (UMLS) (Bodenreider, 2004) (100+), OBO-Foundary (Smith et al., 2007) (~200), Medical Subject Headings (MeSH) (Lipscomb, 2000), Entrez Gene (Maglott et al., 2005), and MEDIC dictionary (Davis et al., 2019) which includes disease entities from MeSH and OMIM (Hamosh et al., 2005).

- **Dataset** - `bioel.datasets` efficiently handle annotated corpora essential for training and evaluating entity linking models. The BioEL dataset object manages datasets from BigBIO (Fries et al., 2022), a resource for biomedical text mining. It supports loading, preprocessing, and splitting datasets into training, validation, and test sets. It utilizes Ab3P (Sohn et al., 2008) to identify and resolve abbreviations at train/inference time.

- **BioEL model** - `bioel.model` provides a unified interface for training and deploying various entity-linking models, supporting model configuration, training, and inference.

- **Evaluation** - `bioel.evaluate` assesses BioEL model outputs by ranking entity candidates for each dataset. Following (Kartchner et al., 2023), it employs metrics like recall@k for $k \in \mathbb{N}$. By default, it uses the basic evaluation strategy, with options to choose between basic, strict, and relaxed approaches. Users can also add custom metrics as needed.

The package allows seamless integration of new datasets and models, enabling users to train, evaluate, and compare state-of-the-art models for comprehensive exploration and development.

3.2 Package Structure Overview

Package components collectively facilitate the end-to-end workflow of data handling, model development, training, and evaluation.

3.2.1 Data Template

The data template serves as a foundational blueprint for managing ontologies and datasets within the package, providing a standardized framework for data structuring. In Biomedical Entity Linking, we handle two main data types: *Ontology* and *Dataset*.

For Ontology, BioEL supports a structured format to define entities. The structure of the `BiomedicalOntology` is as follows:

```
class BiomedicalOntology:
    name: str
    types: List[str]
    entities: Dict[str,
        BiomedicalEntity]
    dataset: Optional[str] = None
```

- `name`: Ontology identifier
- `types`: Types in the Ontology
- `entities`: CUI-to-Entity mappings
- `dataset`: Dataset name (optional)

Note that a dataset name is required for the Entrez Gene Ontology. Each entity within the ontology is encapsulated as an instance of the `BiomedicalEntity` class, which is structured as follows:

```
class BiomedicalEntity:
    cui: str
    name: str
    types: List[str]
    aliases: List[str]
    definition: Optional[str] = None
    equivalant_cuis:
        Optional[List[str]] = None
    taxonomy: Optional[str] = None
    metadata: Optional[dict] = None
```

- `cui`: Concept Unique Identifier
- `name`: Entity name
- `types`: Entity types
- `aliases`: Entity aliases
- `definition`: Entity definition (optional)
- `equivalent_cuis`: Equivalent CUIs (optional)
- `taxonomy`: Entity taxonomy (optional)
- `metadata`: Metadata dictionary (optional)

This structure ensures that the ontology and its entities are defined consistently. Users can add a new ontology by implementing its loader function as a class method within the `BiomedicalOntology` class.

BigBIO datasets (Fries et al., 2022) are the primary framework for dataset management due to their widespread usage. For integrating further datasets, we recommend exploring the collections available on HuggingFace.

We provide the `Mention` class to integrate additional datasets while insuring consistency:

```
class Mention:
    cui: str
    start: int
    end: int
    name: str
    types: List[str]
    deabbreviated_text: Optional[str]
        = None
```

- `cui`: Concept Unique Identifier
- `start`: Start index of the mention
- `end`: End index of the mention
- `name`: Mention name
- `types`: Mention types
- `deabbreviated_text`: Deabbreviated mention (optional)

3.2.2 Data Module

The data module uses `LightningDataModule` from the PyTorch Lightning library (Falcon and team, 2019) to manage model-specific data preparation and processing. It is structured as follows:

- `prepare_data` - Handles one-time tasks like downloading datasets or tokenizing large text corpus on a single GPU.
- `setup` - Initializes and partitions datasets into training, validation, and test sets individually on each GPU, if multiple GPUs are used.
- `train_dataloader` - Creates the `DataLoader` for the train data.
- `valid_dataloader` - Creates the `DataLoader` for the validation data.
- `test_dataloader` - Creates the `DataLoader` for the test data.

These methods efficiently handle and distribute data across multiple GPUs (in a multi-GPU setup), optimizing the model training process by maximizing computational resources and accelerating performance.

3.2.3 Model Template

The Model Template serves as a blueprint for training and evaluating models within the package. To incorporate a new model into the library, users should adhere to the specific structure illustrated in Figure 5 in the Appendix.

In `train.py`, the essential training loop resides within the function placeholder `train_model`, while `evaluate.py` houses the fundamental evaluation loop in the function placeholder `evaluate_model`.

3.2.4 Model Object

The `BioEL_Model` class serves as a versatile interface for managing various biomedical entity linking models within a unified framework. The structure is outlined as follows:

- `Initialization` - Sets up the BioEL model by defining key attributes including its name, the selected entity linking model, paths to training and evaluation scripts, and necessary parameters for configuration and execution.
- `Entity Linking Model` - Creates an instance of the model with configured parameters
- `Training` - Initiates model training using the specified training script path
- `Inference` - Executes model evaluation and inference using the designated evaluation script path
- `Configuration Handling` - Configures model with `config.json`

Users can integrate a new model into the package using the following code snippet:

```
from bioel.models.NewModel.model
    import new_model
from bioel.model import BioEL_Model

class newModel(BioEL_Model):
  @classmethod
  def new_model(cls, name, params,
      checkpoint_path=None):
     model=new_model(params)
     train_script_path=
      "bioel/models/NewModel/train.py"
     evaluate_script_path=
      "bioel/models/NewModel/evaluate.py"
     return cls(model, name,
         train_script_path,
         evaluate_script_path, params)
```

3.2.5 Evaluator

The `Evaluate` class in BioEL, accessed through `bioel.evaluate`, is designed for evaluating model performance on specified datasets. It supports different metrics (Recall@k, MAP@k), and includes functionalities for model analysis (failure stage, error per type $\cdots$), comparison, and score visualization across different models. There are currently 3 different evaluation strategies :

- **Basic**: Resolves ties by randomly ordering equally ranked entities.
- **Relaxed**: An entity link is correct if any predicted normalization matches any ground truth.
- **Strict**: A normalization is correct only if all predicted normalizations match the ground truth.

The Evaluator expects model outputs in a specific format where each sample is represented as a dictionary containing at least `db_ids`, `mention_id`, `type` and `candidates` keys, structured as follows :

```
[
  {
    data about the mention 1 ...
    "db_ids": Gold CUI of mention 1,
    "mention_id": "9288106.1",
    "type": type of mention 1,
    "candidates": [
      [candidate 1 and its equivalent cuis],
      [candidate 2 and its equivalent cuis],
      etc...
      ]
  },
  {
    data about the mention 2 ...
    "db_ids": Gold CUI of mention 2,
    "mention_id": "9288106.2",
    "type": type of mention 2,
    "candidates": [
      [candidate 1 and its equivalent cuis],
      [candidate 2 and its equivalent cuis],
      etc...
      ]
  },
etc..
]
```

If using abbreviations in the analysis, replace `"mention_id": "9288106.1"` with `"mention_id": "9288106.1.abbr_resolved"`

4 Available Models and ontologies

BioEL provides unparalleled flexibility with its support for 6 distinct biomedical entity linking models, paired with a vast array of over 300 ontologies and 26 datasets. This creates a massive range of model-ontology-dataset combinations allowing users to tailor their entity linking tasks with

precision, ensuring the system is adaptable to a variety of biomedical challenges. This versatility positions BioEL as a leading tool in biomedical research, enabling users to explore a huge number of approaches to entity linking. Table 2 summarizes the supported models, datasets and ontologies.

4.1 Models

BioEL supports multiple categories of models:

4.1.1 Alias Matching EL

Alias-based EL associates entity mentions with the correct aliases in a KB. This includes exact string matching and computing similarity scores between the mention and a set of candidate aliases. Alias matching methods include SciSpacy (Neumann et al., 2019) and SapBERT (Liu et al., 2021).

4.1.2 Contextualized EL

Contextualized EL leverages transformer (Vaswani et al., 2023) abilities to capture semantic similarities between contextualized mention and entity metadata (Logeswaran et al., 2019). For instance, (Wu et al., 2020) used a pretrained BERT bi-encoder (Devlin et al., 2019) to first generate candidates based on these similarity scores and then a cross-encoder for the final disambiguation step. The BioEL package features two contextualized EL models: KRISSBERT (Zhang et al., 2022) and ArboEL (Agarwal et al., 2022).

4.1.3 Autoregressive EL

Autoregressive model-based EL approach (Cao et al., 2021) uses a generative language model to map textual mentions directly to canonical entity names. This method contrasts with traditional approaches by not relying on predefined indices for entities, allowing for seamless handling of new and unseen data without re-training. This makes it particularly robust in zero-shot settings. The package includes BioBART (Yuan et al., 2022a) and BioGenEL (Yuan et al., 2022b).

5 Usage

This section explores how to utilize the BioEL library's full range of features.

5.1 Installation

BioEL can be installed via pip:

```
pip install bioel
```
[1]

[1] https://pypi.org/project/bioel/

5.2 Loading a Dataset

Load datasets with BioEL's `Dataset` (from `bioel.dataset`). Here's a code snippet for loading the dataset `BC5CDR` (Li et al., 2016) dataset.

```
from bioel.dataset import Dataset

dataset = BigBioDataset(dataset_name =
    "bc5cdr")
```

5.3 Loading Ontology

BioEL supports loading ontologies using the `BiomedicalOntology` class from the `bioel.ontology` module. The functions adhere to a consistent naming convention, `load_<ontology_name>`, where `<ontology_name>` is substituted with the actual ontology name. The code snippet demonstrates loading MEDIC ontology.

```
from bioel.ontology import
    BiomedicalOntology

ontology =
    BiomedicalOntology.load_medic(
    filepath = "path/to/medic.tsv",
    name="medic")
```

5.4 Loading Model

BioEL provides the `BioEL_Model` class from the `bioel.model` module for loading, training and evaluating the models. The code snippet below illustrates the process of loading the KrissBERT model (Zhang et al., 2022).

```
from bioel.model import BioEL_Model

krissbert =
    BioEL_Model.load_krissbert(params)
krissbert.training()
krissbert.evaluation()
```

Here, the `params` argument refers to a configuration file containing details like the dataset name, ontology, model hyperparameters, training settings, and evaluation configurations. Examples of configuration files for each supported model are provided in the GitHub link.

During execution, the `krissbert.evaluation()` method generates a `result.json` file formatted to be suitable for the `Evaluate` object.

5.4.1 Evaluating Results

BioEL facilitates model evaluation through the `Evaluate` class from the `bioel.evaluate` module. The snippet below demonstrates the process:

Models	Supported Ontologies	Datasets
SapBERT, SciSpacy, KRISSBERT, ArboEL, BioBART, BioGenEL	UMLS (**100+**), OBO (Gathers ~**200**), MEDIC, Entrez,	BigBio datasets (**26**)

Table 2: Available Datasets and Models in **BioEL**.

	Impl.	NCBI-Disease		BC5CDR		MM-Full		MM-ST21PV		GNormPlus		NLM-Chem		NLM-Gene	
		1	5	1	5	1	5	1	5	1	5	1	5	1	5
SapBERT		0.753	0.899	0.873	0.926	0.608	0.795	0.594	0.771	0.174	0.538	0.755	0.876	0.066	0.314
		0.753	0.896	0.883	0.934	0.611	0.786	0.637	0.788	0.234	0.614	0.812	0.889	0.075	0.348
KRISSBERT		0.745	0.797	0.720	0.757	0.583	0.745	0.548	0.689	0.108	0.121	0.551	0.583	0.280	0.484
		0.752	0.803	0.735	0.766	0.591	0.755	0.559	0.701	0.079	0.087	0.560	0.596	0.279	0.482
SciSpacy		0.697	0.838	0.843	0.902	0.590	0.773	0.577	0.751	0.110	0.375	0.616	0.706	0.055	0.228
		0.680	0.780	0.780	0.803	0.582	0.759	0.572	0.741	0.471	0.772	0.467	0.503	0.163	0.349
ArboEL		0.771	0.820	0.902	0.938	NR	NR	0.687	0.798	0.585	0.647	0.790	0.857	0.560	0.751
		0.774	0.832	0.921	0.958	NR	NR	0.747	0.890	0.441	0.524	0.828	0.882	0.543	0.734
BioBART		0.728	0.834	0.864	0.921	0.586	0.805	0.569	0.788	0.112	0.363	0.721	0.823	0.061	0.319
		0.423	0.608	0.572	0.733	0.548	0.764	0.496	0.700	0.175	0.499	0.512	0.650	0.043	0.233
BioGenEL		0.734	0.851	0.867	0.920	0.574	0.783	0.561	0.764	0.141	0.412	0.743	0.856	0.062	0.305
		0.518	0.692	0.909	0.953	0.567	0.763	0.520	0.691	0.081	0.281	0.786	0.879	0.043	0.233

Table 3: Comparison Recall@1 and recall @ 5 between (Kartchner et al., 2023) and replicated performance for models in the BioEL Python package. NR=Not reproducible due to computational constraints.

```python
from bioel.evaluate import Evaluate

eval_strategies = ["basic"]
dataset_names = ["bc5cdr"]
model_names = ["krissbert"]
path_to_result = {
  "bc5cdr": {
    "krissbert": "path/to/result.json"
  }
}
eval_strategies=["basic"]
abbreviations_path =
    "path/to/abbreviations.json"

evaluator = Evaluate(dataset_names,
    model_names, path_to_result,
    eval_strategies,
    abbreviations_path)
evaluator.load_results()
evaluator.process_datasets()
evaluator.evaluate()
evaluator.plot_results()
evaluator.detailed_results()
```

- `datasets`: List of dataset names to be evaluated
- `models`: List of model names to be evaluated
- `results_path`: directory where `result.json` files each model are stored.
- eval_strategy: "basic", "relaxed", "strict"
- `abbreviations_path`: (Optional) Path to JSON file containing mappings for abbreviations.

5.4.2 Evaluation features

The `Evaluate` class from the `bioel.evaluate` module supports a variety of statistical tests, including fine-grained metrics computation through the `detailed_results_analysis` attribute. It provides stratified evaluation per entity type (e.g., chemical vs. disease), which allows users to gain deeper insights into model performance across different semantic categories. This process determines whether the observed variations in metrics, such as accuracy, are statistically significant, thereby enabling a more rigorous and comprehensive evaluation of model performance.

5.5 Documentation

Additional resources on how to modify the datasets, the ontologies, and how to utilize the different models, as well as how to initiate their training, perform inference, and review all evaluation results, are provided in the README.md file available in the GitHub repository.

6 Results

BioEL was benchmarked to ensure the BioEL Python package re-implementation accurately replicated the included models.

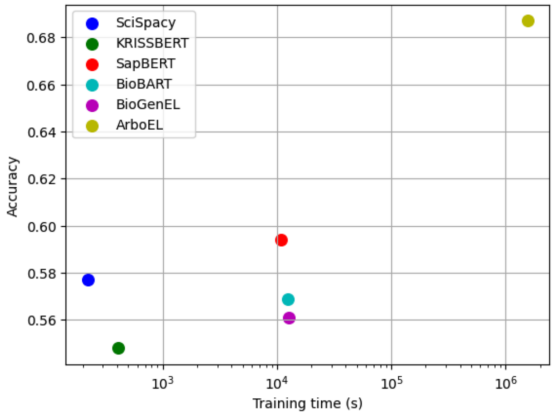

Figure 3: Accuracy vs Training time (s) for all models supported by BioEL on MM-ST21PV dataset

6.1 Main Results

The methodology of (Kartchner et al., 2023) was closely followed. The main findings, detailed in Table 3, show recall@1 (accuracy) and recall@5 for all models included in the package across various datasets. Figure 4 provides a visual illustration of the changes in recall@k for $k = 1, \ldots, 10$, generated by the BioEL evaluator object. The training time for all models supported by BioEL was reported for the largest dataset, MM-ST21PV, as shown in Figure 3. All experiments were conducted on a single Nvidia A40 GPU.

Additional results were included in the appendix, such as failure stage analysis (Table 7), performance by entity type (Table 4), statistical significance (Table tests 5), and Mean Average Precision (MAP) (Table 6).

6.2 Difference in reported results

For most models and datasets, BioEL results closely aligned with those reported by (Kartchner et al., 2023). Notably, KRISSBERT required minimal modifications, as we used the model with its original pre-trained weights, ensuring strong consistency. In contrast, models that required training or fine-tuning, along with adjustments to data processing for better generalization across different datasets, occasionally exhibited discrepancies. These differences could be attributed, for example, to variations in hyperparameters.

6.2.1 ArboEL

ArboEL requires both the title and description of entities. In this package, we standardized the description pattern across all KBs to enhance generalizability as mentioned in 2.2.1. This standardization may account for the minor differences shown.

6.2.2 Scispacy

In (Kartchner et al., 2023) work, the Approximate Nearest Neighbor (ANN) was trained exclusively on UMLS across all datasets. To derive their results, they remapped the CUIs from UMLS to those of their respective original ontologies, which is not the optimal methodology for entity linking. For the BioEL evaluation, we aligned the training of the ANN with the specific ontology relevant to each dataset. This nuanced approach explains the similar performance for datasets using UMLS and the superior performance for datasets employing the MEDIC and MeSH ontologies (due to a reduced number of aliases). Conversely, the outcomes for the GNormPlus and NLMGene datasets declined because the initial UMLS encompasses only a limited subset of the most frequent CUIs from Entrez Gene, drastically narrowing the spectrum of plausible aliases and therefore candidates; making the task simpler.

6.2.3 BioGenEL / BioBART

In the BioEL benchmark, both BioGenEL and BioBART were fine-tuned using their respective pretrained weights available in the original BioGenEL repository and on HuggingFace. These models were fine-tuned with the same trainer, utilizing the BART-large tokenizer to generate the results. Note that the outcomes are within the range of the results of (Kartchner et al., 2023) and those reported by the original authors (Yuan et al., 2022a). In the BioEL package, entity definitions were intentionally omitted while creating the target knowledge base (KB) for the Entrez ontology, as only the aliases are required. This omission accounts for the differences in results for the GNormPlus and NLMGene datasets. Additional variances are anticipated due to changes in data preprocessing, such as tokenized inputs for the encoder and decoder, optimization techniques, and other user-specific parameter selections.

6.2.4 SapBERT

The main source of variance arises from differences in the fine-tuning strategies. While we fine-tuned SapBERT on UMLS combined with its associated ontology, the authors of the referenced paper fine-tuned it directly on the associated ontology.

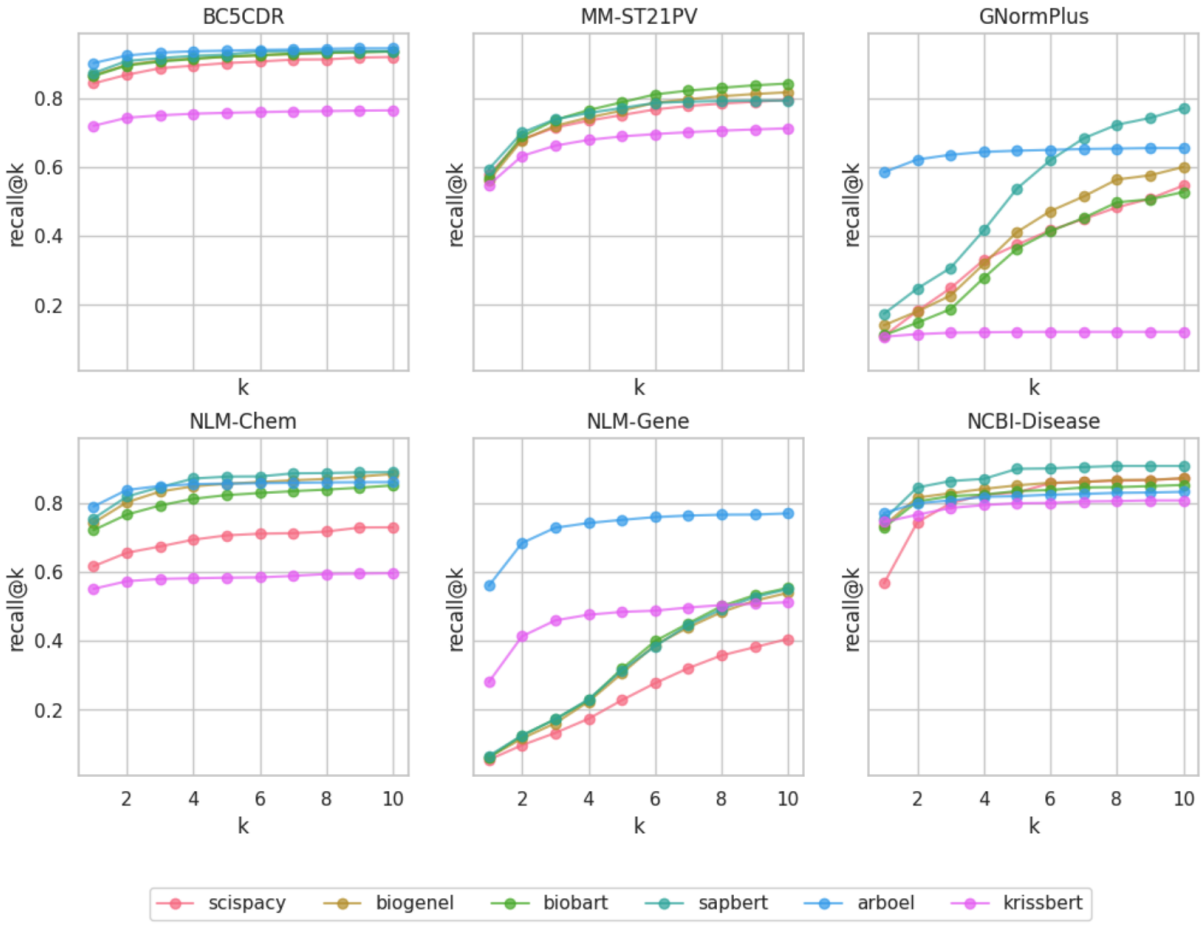

Figure 4: Recall@K for all models using basic evaluation strategy (generated by BioEL Evaluator)

7 Conclusion

BioEL is a free, open-source Python library designed to simplify biomedical entity linking. It features a user-friendly interface and flexible design, enabling researchers to easily apply advanced entity-linking techniques. BioEL offers robust tools for ontology management, dataset handling, and model training and evaluation, supporting various models with comprehensive metrics. Its extensibility and configuration options allow for the customization and integration of new models or datasets. Benchmarking demonstrates BioEL's effectiveness across diverse datasets, enhancing information extraction and integration from the scientific literature to advance biomedical research.

8 Limitations and Future Directions

While BioEL provides a convenient and integrated platform for conducting EL analysis, supporting reproducible research, and fostering future advancements, it also has inherent limitations. Its association with the biomedical domain raises concerns about potential misuse, especially in healthcare settings. To mitigate these risks, we recommend users refer to comprehensive discussions and guidelines on responsible AI in biomedical research (Blasimme and Vayena, 2020).

Currently, BioEL incorporates versatile models, yet lacks specialized models tailored for specific entities such as genes and chemicals, which could enhance performance on specific datasets. Additionally, it's crucial to acknowledge the evolution of knowledge bases. Therefore, we strongly advise users to thoroughly evaluate both the models and the overall BioEL system before proceeding with practical implementations.

Priorities for future releases of BioEL include more entity-linking models and a wider range of datasets that go beyond the currently supported BigBio datasets. Contributions from the open-source community are encouraged to join this evolving effort.

Acknowledgments

Funding support provided by National Science Foundation CAREER grant 1944247, National Institute of Health grant R35GM152245, and Chan Zuckerberg Initiative grant 253558 to C.S.M.

References

Dhruv Agarwal, Rico Angell, Nicholas Monath, and Andrew McCallum. 2022. Entity linking via explicit mention-mention coreference modeling. pages 4644–4658.

Alessandro Blasimme and Effy Vayena. 2020. The ethics of ai in biomedical research, patient care, and public health. In Markus D. Dubber, Frank Pasquale, and Sunit Das, editors, *The Oxford Handbook of Ethics of AI*. Oxford Academic. Online edition, 9 July 2020.

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

Florian Borchert, Ignacio Llorca, and Matthieu-P Schapranow. 2024. Improving biomedical entity linking for complex entity mentions with LLM-based text simplification. *Database*, 2024:baae067.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Haihua Chen, Ruochi Li, Junhua Ding, and Ana Cleveland. 2024. Enhancing data quality in medical concept normalization through large language models. *SSRN*. Available at SSRN: https://ssrn.com/abstract=4979696 or http://dx.doi.org/10.2139/ssrn.4979696.

Allan Peter Davis, Cynthia J Grondin, Robin J Johnson, Daniela Sciaky, Roy McMorran, Jolene Wiegers, Thomas C Wiegers, and Carolyn J Mattingly. 2019. The comparative toxicogenomics database: update 2019. *Nucleic acids research*, 47(D1):D948–D954.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186.

William Falcon and The PyTorch Lightning team. 2019. Pytorch lightning. *arXiv preprint arXiv:1910.10683*.

Jason A. Fries, Troy Mayfield, Kenneth Brimacombe, Michael M. Bronstein, David Silver, David Wehner, et al. 2022. Bigbio: A large-scale biomedical corpus. *Journal of Biomedical Informatics*, 135:104037.

Ada Hamosh, Alan F Scott, Joanna S Amberger, Carol A Bocchini, and Victor A McKusick. 2005. Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders. *Nucleic acids research*, 33(suppl_1):D514–D517.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

David Kartchner, Jennifer Deng, Shubham Lohiya, Tejasri Kopparthi, Prasanth Bathala, Daniel Domingo-Fernández, and Cassie Mitchell. 2023. A comprehensive evaluation of biomedical entity linking models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14462–14478, Singapore. Association for Computational Linguistics.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016:baw068.

Carolyn E Lipscomb. 2000. Medical subject headings (mesh). *Bulletin of the Medical Library Association*, 88(3):265.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2021. Self-alignment pretraining for biomedical entity representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4228–4238, Online. Association for Computational Linguistics.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. pages 3449–3460.

Daniel Loureiro and Alípio Mário Jorge. 2020. Medlinker: Medical entity linking with neural representations and dictionary matching. *Advances in Information Retrieval*, 12036:230 – 237.

Donna Maglott, Jim Ostell, Kim D Pruitt, and Tatiana Tatusova. 2005. Entrez gene: gene-centered information at ncbi. *Nucleic acids research*, 33(suppl_1):D54–D58.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics.

Pedro Ruas and Francisco M. Couto. 2022. Nilinker: Attention-based approach to nil entity linking. *Journal of Biomedical Informatics*, 132:104137.

Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J Mungall, et al. 2007. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, 25(11):1251–1255.

Sunghwan Sohn, Donald C Comeau, Won Kim, and W John Wilbur. 2008. Abbreviation definition identification based on automatic precision estimates. *BMC bioinformatics*, 9(1):1–10.

Mujeen Sung, Hwisang Jeon, Jinhyuk Lee, and Jaewoo Kang. 2020. Biomedical entity representations with synonym marginalization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3641–3650, Online. Association for Computational Linguistics.

Mujeen Sung, Minbyul Jeong, Yonghwa Choi, Donghyeon Kim, Jinhyuk Lee, and Jaewoo Kang. 2022. Bern2: an advanced neural biomedical namedentity recognition and normalization tool.

Mario Sänger, Samuele Garda, Xing David Wang, Leon Weber-Genzel, Pia Droop, Benedikt Fuchs, Alan Akbik, and Ulf Leser. 2024. Hunflair2 in a cross-corpus evaluation of biomedical named entity recognition and normalization tools. *Bioinformatics*, 40(10).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Brian Walsh, Sameh K. Mohamed, and Vít Nováček. 2020. Biokg: A knowledge graph for relational learning on biological data. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 3173–3180, New York, NY, USA. Association for Computing Machinery.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval.

Hongyi Yuan, Zheng Yuan, Ruyi Gan, Jiaxing Zhang, Yutao Xie, and Sheng Yu. 2022a. BioBART: Pretraining and evaluation of a biomedical generative language model. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 97–109, Dublin, Ireland. Association for Computational Linguistics.

Hongyi Yuan, Zheng Yuan, and Sheng Yu. 2022b. Generative biomedical entity linking via knowledge base-guided pre-training and synonyms-aware fine-tuning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4038–4048, Seattle, United States. Association for Computational Linguistics.

Sheng Zhang, Hao Cheng, Shikhar Vashishth, Cliff Wong, Jinfeng Xiao, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2022. Knowledge-rich self-supervision for biomedical entity linking.

A Appendix

We organize the appendix into three sections:

- Further details on the Model Template are provided in Appendix A.1.
- Additional results and analysis can be found in Appendix A.2.
- Discussions on the potential use of LLMs for entity linking are presented in Appendix A.3.

A.1 Model Template

Figure 5 shows the structure users must follow to add their own model to BioEL.

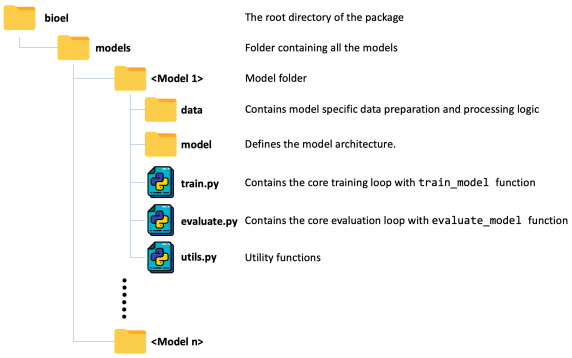

Figure 5: Model structure

The diverse nuances and variances in biomedical entity linking models necessitated this specific design of the BioEL architecture. To ensure consistency and reproducibility with the original works, each model uses its own training and evaluation script, allowing seamless integration of original code while minimizing new variances. Importantly, this design maintains ease of use and supports future expansion to new models.

A.2 Additional Results

In this section, we present additional results, including evaluation by entity type, MAP@k scores, and an error analysis detailing failure stages (CG or NED).

A.2.1 Evaluation per entity type for MM-ST21PV

In Table 4, we show the performance recall@1 and recall@5 on the different types within MedMentions_st21pv dataset for all supported models.

	ArboEL		BioBART		BioGenel		KRISSBERT		SapBERT		SciSpacy	
	1	5	1	5	1	5	1	5	1	5	1	5
Virus	0.703	0.808	0.663	0.930	0.663	0.895	0.465	0.564	0.709	0.878	0.634	0.883
Bacterium	0.693	0.766	0.621	0.759	0.633	0.797	0.452	0.535	0.624	0.751	0.590	0.751
Anatomical Structure	0.659	0.748	0.610	0.793	0.584	0.768	0.515	0.639	0.637	0.772	0.619	0.763
Body System	0.767	0.811	0.578	0.811	0.622	0.833	0.689	0.856	0.656	0.800	0.589	0.800
Body Substance	0.830	0.892	0.802	0.910	0.793	0.877	0.689	0.807	0.745	0.882	0.745	0.816
Finding	0.703	0.851	0.545	0.755	0.546	0.737	0.558	0.701	0.572	0.776	0.546	0.723
Injury or Poisoning	0.647	0.849	0.493	0.756	0.518	0.784	0.476	0.650	0.580	0.776	0.504	0.748
Biologic Function	0.741	0.851	0.626	0.858	0.613	0.833	0.592	0.731	0.674	0.839	0.660	0.817
Health Care Activity	0.645	0.784	0.448	0.758	0.460	0.746	0.517	0.691	0.489	0.762	0.474	0.739
Research Activity	0.800	0.902	0.393	0.632	0.380	0.603	0.722	0.884	0.422	0.612	0.400	0.581
Medical Device	0.544	0.710	0.530	0.699	0.467	0.631	0.260	0.431	0.501	0.606	0.515	0.654
Spatial Concept	0.750	0.844	0.656	0.820	0.633	0.798	0.659	0.800	0.671	0.798	0.678	0.809
Biomedical Occupation or Discipline	0.740	0.867	0.755	0.878	0.744	0.883	0.541	0.791	0.776	0.893	0.755	0.888
Organization	0.649	0.764	0.602	0.751	0.576	0.738	0.547	0.691	0.586	0.673	0.605	0.691
Professional or Occupational Group	0.731	0.864	0.689	0.838	0.686	0.800	0.631	0.775	0.700	0.839	0.675	0.828
Population Group	0.862	0.972	0.580	0.885	0.572	0.865	0.801	0.935	0.615	0.867	0.584	0.847
Chemical	0.594	0.679	0.575	0.758	0.571	0.729	0.429	0.552	0.603	0.752	0.580	0.717
Food	0.630	0.736	0.693	0.817	0.618	0.742	0.497	0.668	0.559	0.646	0.568	0.668
Intellectual Product	0.647	0.792	0.482	0.714	0.433	0.659	0.540	0.711	0.445	0.638	0.444	0.638
Clinical Attribute	0.858	0.913	0.483	0.848	0.474	0.882	0.752	0.824	0.678	0.879	0.511	0.848
Eukaryote	0.694	0.789	0.647	0.844	0.635	0.823	0.511	0.627	0.617	0.805	0.625	0.806

Table 4: Performance recall@1 and recall@5 on the different types within MedMentions_st21pv dataset for all supported models.

A.2.2 Statistical Significance Tests

In Table 5, we provide the p-values for the accuracy of all evaluated models and datasets. The p-values determine whether the performance differences across various types within a dataset are statistically significant.

MM-Full and MM-ST21PV consistently show extremely low p-values across all models, indicating that there is a significant statistical difference in performance for all models on these datasets. All models exhibit varying performance on different data types for these datasets.

On the other hand, for datasets such as BC5CDR and NLM-Gene, some models show higher p-values (e.g., ArboEL with a p-value of 0.812 on BC5CDR or BioBART with 0.353 on NLM-Gene). These higher p-values indicate that these models are more robust across the different types for these datasets.

A.2.3 Mean Average Precision

In Table 6, we show the score of MAP@1 and MAP@5 for all evaluated models and datasets.

A.2.4 Failure Stage

To compute the scores for Candidate Generation (CG) and Named Entity Disambiguation (NED) failures in 7, we defined a CG failure as occurring when the correct CUI does not appear within the top-k candidates generated. For NED failures, we only counted instances where the top-ranked candidate was incorrect, given that the correct CUI was still present in the list of plausible candidates.

A.3 Potential role of LLM for entity-linking

Currently, LLMs are not ideal for entity linking due to several key limitations. Firstly, they are not inherently aware of the databases in the different ontologies, which often leads to hallucination when generating candidates even with explicit information. This makes them completely unreliable for the candidate generation step. While fine-tuning them could mitigate this issue, it is both more resource-intensive and challenging given their size, especially for domain-specific tasks like biomedical entity linking.

However, people are leveraging LLMs as an external tool to enhance the data quality for traditional biomedical entity linking models (Borchert et al., 2024) (Chen et al., 2024).

Regardless of the LLM usage and performance, this package offers researchers an additional tool designed to support their work in biomedical entity linking.

	NCBI-Disease	BC5CDR	MM-Full	MM-ST21PV	NLM-Gene
SapBERT	5.18^{-6}	4.45^{-88}	0.00	6.75^{-216}	3.62^{-2}
KRISSBERT	1.81^{-8}	2.17^{-20}	0.00	1.74^{-293}	3.56^{-8}
SciSpacy	1.69^{-3}	2.66^{-126}	0.00	2.05^{-217}	1.29^{-1}
ArboEL	7.89^{-4}	8.12^{-1}	NR	4.86^{-191}	7.14^{-5}
BioBART	8.09^{-6}	1.32^{-1}	0.00	2.26^{-199}	3.53^{-1}
BioGenEL	3.17^{-6}	7.43^{-3}	0.00	2.67^{-201}	9.87^{-3}

Table 5: P-value of recall@1 for all evaluated models and datasets. GNormPlus and NLM-Chem were not shown because they only have 1 type. NR=Not reproducible due to computational constraints.

	NCBI-Disease		BC5CDR		MM-Full		MM-ST21PV		GNormPlus		NLM-Chem		NLM-Gene	
	1	5	1	5	1	5	1	5	1	5	1	5		
SapBERT	0.752	0.812	0.873	0.895	0.608	0.686	0.594	0.667	0.174	0.282	0.755	0.803	0.066	0.142
KRISSBERT	0.745	0.765	0.720	0.735	0.583	0.650	0.548	0.606	0.108	0.113	0.551	0.565	0.280	0.368
SciSpacy	0.567	0.681	0.843	0.865	0.590	0.665	0.577	0.649	0.110	0.197	0.616	0.649	0.055	0.109
ArboEL	0.771	0.791	0.902	0.917	NR	NR	0.687	0.731	0.585	0.611	0.790	0.819	0.560	0.642
BioBART	0.728	0.774	0.864	0.886	0.586	0.674	0.569	0.656	0.112	0.183	0.721	0.760	0.061	0.141
BioGenEL	0.734	0.784	0.867	0.888	0.574	0.659	0.561	0.644	0.141	0.218	0.742	0.788	0.062	0.136

Table 6: MAP@1 and MAP@5 for all evaluated models and datasets. NR=Not reproducible due to computational constraints.

	NCBI-Disease		BC5CDR		MM-Full		MM-ST21PV		GNormPlus		NLM-Chem		NLM-Gene	
	CG	NED	CG	NED	CG	NED	CG	NED	CG	NED	CG	NED	CG	NED
SapBERT	0.307	0.693	0.501	0.499	0.469	0.531	0.508	0.492	0.182	0.818	0.451	0.549	0.196	0.804
KRISSBERT	0.686	0.314	0.800	0.200	0.450	0.550	0.547	0.453	0.983	0.017	0.859	0.141	0.638	0.362
SciSpacy	0.252	0.748	0.513	0.487	0.464	0.536	0.486	0.514	0.510	0.490	0.706	0.294	0.629	0.371
ArboEL	0.664	0.336	0.500	0.500	NR	NR	0.591	0.409	0.817	0.183	0.653	0.347	0.493	0.507
BioBART	0.437	0.563	0.428	0.572	0.284	0.716	0.283	0.717	0.326	0.674	0.442	0.558	0.135	0.865
BioGenEL	0.357	0. 643	0.402	0.598	0.321	0.679	0.327	0.673	0.224	0.776	0.345	0.655	0.130	0.870

Table 7: Percentage of failure stage for each evaluated model and dataset. NR=Not reproducible due to computational constraints.